Discourse Segmentation in Aid of Document Summarization

Branimir K. Boguraev and Mary S. Neff
IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598
bkb@watson.ibm.com, neff@watson.ibm.com

Abstract

This paper describes work to enhance a sentence-based summarizer with notions of salience, dynamically-adjustable summary size, discourse segmentation, and awareness of topic shifts. Our experiments study strategies to diversify the application of a baseline summarizer, by making it aware of finer-grained 'aboutness', capable of discerning changes of topic, and sensitive to longer-than-usual documents. Evaluated against the corpus used in the development of the baseline summarizer, summaries derived either by means of segmentation analysis alone, or by a mix of strategies for combining salience calculation and topic shift detection, are shown to be of comparable, and under certain conditions even better, quality. We describe the summarization and segmentation procedures, outline a number of strategies for mixing the two, evaluate the overall impact of discourse segmentation, and suggest an interface design capable of using the notion of topic shifts to contextualize a summary and facilitate the mediation between it and the full document source.

1. Introduction

Document summarization has become *de facto* a critical component in any toolkit for on-line information management, as witnessed at least by dedicated conferences and symposia [1], coordinated evaluation initiatives [12], and real-world deployment [7]. Still, in the absence of a coherent theory of summarization, and even less so of a formal computational model of summary derivation, virtually all general purpose summarizers (whether in wide deployment, or of a more experimental nature) currently use variations on the same theme: they compose a summary by 'stitching together' representative fragments—typically sentences—from the original full length document text.

This strategy is sub-optimal, as users have to contend with *loss of coherence*, *deterioration of readability*, and *thematic under-representation* [4]. To a large extent all of these problems stem from arbitarily long passages of the original document being omitted between any two adjacent sentences in the summary; thus loss of essential information[1] interferes with the intended use of the summary.

Even if users are prepared to compromise, in order to get some idea of what a document is about without having to read all of it, such factors lead to rapid degradation of the usefulness of a sentence-based summary in situations beyond the most typical "what is this news story about". Examples of such situations might include: occasions when traditional methods are applied to documents larger than a couple-of-pages-long news article; when the user needs more complete awareness of all major themes in a document; or when different summaries might be appropriate to different user information-seeking contexts.

We have chosen to address such problems by enhancing a sentence-based summarizer with notions of *salience* (as determined with respect to a background document collection) and *dynamically-adjustable size* of the resulting summaries (see [25], and below). However, by focusing on salience as a solution to one set of problems, we become dependent on statistics of a background collection, which clearly limits the applicability of the summarizer across a range of document types and genres.[2] Furthermore, it is far from clear that salience alone offers a complete solution to the problems of incoherence and thematic underrepresentation: for instance, it is not clear how to use it in environments where it is essential to track all the topics/sub-stories in the original document, or to remain sensitive to changing user profiles and interests.

This paper describes some early work on leveraging elements of the larger discourse structure in an attempt to enhance the operation of a salience-based sentence extraction summarizer. In the longer term, this is just one aspect of a larger study on the recognition and use of cohesive devices for a variety of content characterisation tasks. As such, it presupposes fine-grained methods for the identification of cohesive ties between (sentence) units in a text; such ties are typically manifested in textual substitution, lexical repe-

[1] For instance: a "dangling" anaphor, without an antecedents; the reversal of a core premise in an argument; the introduction, and/or elaboration, of a new topic—these are just a few examples of missing essentials.

[2] It it is possible to supply a 'generic' background collection, against which summaries could be generated even for documents which are not *a priori* part of the collection. This is problematic, at least because it is a highly genre-dependent approach. In addition, the generation of a background collection and statistics for it might be impractical for a variety of reasons: lack of access to a sufficiently large and representative data sample; no time for processing; sparse storage resources; and so forth.

tition, co-reference and ellipsis, paraphrasing, conjunction, and so forth. Even if a framework for such analysis takes a while to implement, in the immediate term a 'working approximation' is provided by the phenomenon of simple lexical repetition. We use this to develop an operational definition of discourse segmentation, where segments in a document are defined to be contiguous blocks of text (typically spanning several paragraphs), roughly 'about the same thing', with segment boundaries indicative of topic shifts, and/or changes in themes of discussion.

1.1. Segmentation-assisted summarization

In our work on enhancing summarization by folding in results of linear discourse segmentation, we appeal to a number of common intuitions. In general, we focus on strategies to diversify a summarizer, by making it aware of finer-grained 'aboutness', capable of discerning topic shifts, and sensitive to longer-than-usual documents. In a sentence extraction-based model of summarization, making certain that a summary incorporates sentences from each segment seeks to ensure uniform representation of all sub-stories in a document; the notion here is to avoid having inordinately large gaps between two adjacent summary sentences, which would tend to lose essential information. Moreover, assuming a mechanism which would pick the sentence(s) within a segment which are representative of the main topic discussed in the segment, such a selection strategy would carry over into the summary 'traces' of *all the main topics* in the original document.

This is more than just an intuition. In the process of developing, and training, the base summarization function described below (Section 2.2), an analysis was carried out to determine the causes of a certain class of failure.[3] It turns out that 30.7% of the failures could be prevented by a heuristic sensitive to the logical structure of documents, which would enforce that each section gets represented in the summary. Additional 15.2% of failures could also be avoided if the summarizer was capable of detecting sub-stories within a single section, leading/trailing noise (see below), and so forth. Thus almost half of the errors (in this particular task, at least) could have been avoided by using a segmentation component.

The specific strategies for being sensitive to foci of attention within a segment, and topic shifts between segments, may vary, depending on other environment settings for the summarizer; we return to this question below (Section 3). As we shall see, even very simple approaches—say, take the first sentence from each segment—have remarkably noticeable impact in certain situations.

While segmentation offers plausible schemes for deriving sentence-based summaries with certain discourse properties, it turns out that at least one such scheme also allows the summarizer to operate—in certain cases very effectively—without a need for background corpus statistics.

Another use for a segmentation component in summarization context is for optimising the use of source input, as well as possibly maximising its re-use. Occasionally, the document contains 'noise'—this may be in the form of *anecdotal leads*, *closing remarks* tangential to the main points of the story, *side-bars*, and so forth—which should not be considered as source for summary sentences. Linear segmentation sensitive to topic shifts and document structure would identify such source fragments and remove them from consideration by the summarizer. Conversely, in certain genres of news reporting a whole document fragment (typically towards the beginning or the end of the document) functions as a summary of the story: we would like to be able to use this fragment; clearly identifying it as a segment is part of the whole task.

We also use segmentation to handle long documents more effectively. While the collection-based salience determination works reasonably well for the average-length news story, it has some disadvantages. For longer documents, with requisite longer summaries, the notion of salience degenerates, and the summary takes on more of the appearance of an incoherent collection of sentences. In certain contexts, paragraph-, rather than sentence extraction, has been proposed as a working solution; see e.g. [37]. Apart from inherently suited for longer texts, due to its larger granularity, this suffers from the same problems of patchiness and/or under-representation brought up earlier in this section [23]. We use segmentation to identify contiguous sub-stories in long documents, which are then individually passed on to the summarizer; the results of sub-story summaries are 'glued' together.

The remainder of this paper is organized as follows. Section 2 presents an overview of the document processing infrasructure within which the summarization function is just one component, and gives some details about the processes of summary generation and linear discourse segmentation. We focus in particular on how the higher level content analysis functions make use of lower level shallow linguistic processing, in order to obtain a richer model of the document(s) domain, and to leverage a cohesion metric for sub-story identification. Section 3 presents the results from a number of experiments, comparing the performance of summarization alone to segmentation-enhanced summarization; to set the context, we outline the evaluation testbed environment we use. Following a discussion of the results, which suggest specific run-time strategies for optimally using the notions of discourse segments and topic shifts for summarization, we outline some core features of an interface which tries to make 'visual sense' of the notions we use (salience, topics, summary sentences, discourse segments, context, and so forth). We conclude with an assessment of the overall utility of 'cheap' approximations to lexical coherence measures, specifically from the point of view of enhancing a fully operational summarizer engine.

[3] In a task-based evaluation protocol (see Section 3.1 below), quality of summaries was assessed by using them to determine whether a document is relevant to a query or not. The evaluation environment provided a training corpus, against which the summarizer was developed, and which was used as the basis for our analysis.

2. Background technologies

Unlike most operational summarization systems to date, the one discussed here is an integral component of a much larger infrastructure for document processing and analysis, comprising a number of interconnected, and mutually enabling, linguistic filters. The whole infrastructure (hereafter referred to as TEXTRACT) is designed from the ground up to perform a variety of linguistic feature extraction functions, ranging from straightforward, single pass, tokenisation, lexical look-up and morphological analysis, to complex aggregation of representative (salient) phrasal units across large multi-document collections. To a large extent these characteristics of our document processing environment define the basic design decisions concerning the specifics of our summarizer: sentence selection based upon salience ranking of phrasal units in individual documents, against a background of the distribution of phrasal vocabulary across a large multi-document collection.

2.1. Textract infrastructure

For the purposes of this paper, TEXTRACT can be viewed as a robust text analysis system that identifies proper names and technical terms, along with their variants (contractions, abbreviations, colloquial uses, and so forth) in individual documents in a multi-document collection, and builds a collection vocabulary of canonical forms and variants with statistical information concerning their distribution behaviour and prominence patterns across the collection. The collection vocabulary and statistics are used in the summarizer's salience calculation, which, in turn, is a significant component of the sentence-level score that selects the sentences for extraction.

Most of the linguistic analysis of TEXTRACT utilized by the summarizer is derived through a variety of shallow techniques. This is partly motivated by the requirements of an operational and robust system capable of efficient processing of thousands of documents/gigabytes of data. Alternatively, this can be viewed as an ongoing investigation into how much of higher level semantic and discourse functions can be realized from a shallow linguistic base [18]. In any case, we disagree with claims that morphological analysis and multi-word identification would complicate processing, without benefit to function (see, for instance, [5]). The TEXTRACT system, known commercially as *Intelligent Miner for Text*, is an IBM product which has been successfully deployed in a number of operational information management environments (see, for instance, [6], [27]); its summarizer component is comparable in performance to other industry-strength state-of-the-art technologies [21].

As a fundamentally frequency-based system, the summarizer is ideally positioned to exploit TEXTRACT's functions for linguistic analysis, filtering, and normalization. Thus, morphological processing allows us to link multiple variants of the same word, by normalizing to lemma forms. A proper name identifier, NOMINATOR, [34] not only marks *"Bill"* as a `name⇒person`, but also distinguishes between it and *"bill"*, thus reducing noise in the frequency counting [39]. Further, its ability to identify *"Bill Clinton"* and *"Clinton"* as variants of the same name boosts the frequency of the concept (and ultimately its salience) in the document. Similarly, a light-weight component for resolving definite noun phrase anaphora identifies *"the law firm"* and later *"the firm"* as co-referring, allowing both to be counted together. The interaction of NOMINATOR with ABBREVIATOR makes it possible to recognize *"American Bar Association"* and its variant *"ABA"* as also co-referring. Yet a different component, TERMINATOR, implements a version of technical terminology identification and extraction [16]; this enables the recognition of certain multi-word concepts mentioned in the document, with discourse properties which reflect high topicality value, which is also directly relevant to salience determination. The interaction of NOMINATOR with TERMINATOR makes it possible to analyze *"Treasury bill"* and *"Alzheimer's disease"* as multi-word phrasal units.

In the analysis of a multi-document collection, each document is analyzed individually. All 'content' words (non-stop words, in Information Retrieval terminology), as well as all the phrasal units identified by the TEXTRACT linguistic filters, are deemed to be *vocabulary items*, indexed via their canonical forms. With a view to future extensions of the base summarization function (see Section 5), these retain complete contextual information about the variants in which they have been encountered, as well as the local context of each occurrence. The vocabulary items are counted and aggregated across documents to form the *collection vocabulary*. Aggregating together similar items from different documents (cross-document co-reference) is far from straightforward for multi-word items; however, being able to carry out a process of cross-document coreference resolution is clearly a further enabling capability for obtaining more precise collection statistics [33].

In addition to the domain vocabulary, the summarizer also has access to the *document structure* provided by the TEXTRACT base. The document structure builder produces a structural representation of the document, which carries explicit identification of content and layout metadata. These include: appearance and layout tags; document title; abstract, and other front matter; section, subsection, etc. headings; paragraphs, themselves composed of sentences; tables, figures, captions, and other 'floating' objects; sidebars and other kinds of text extraneous to the main document narrative; and so forth. At present, document structure is constructed by 'shadowing' markup parsing, as markup tags are used to construct the document structure tree. For documents which lack markup tags, a separate component, LAYSER (LAYout parSER), facilitates the document structure builder by carrying out structure determination on the basis of two-dimensional (page) layout cues. Additional discourse-level annotations may also be recorded in the document structure, such as cue phrases marking rhetorical relations, quoted speech, and so forth.

2.2. Summarization component

The TEXTRACT summarizer was explicitly designed to leverage TEXTRACT's linguistic filters for the analysis of documents. It is a frequency-based system; however, due to the depth of analysis by the filters (see Section 2.1), it is able to exploit a richer source of domain knowledge than most other frequency-based systems. We are not alone in exploiting linguistic dimensions beyond single word analysis (see [2], for instance, for a sentence-based summarizer using multi-word sequences). The motivation for such an approach—intuitively, lack of discourse processing adversely affects the quality of an abstract—has been formulated a while ago [29], and reiterated since [15], [28]; but it is only recently that robust shallow and scalable techniques have been developed for unconstrained texts.

Early frequency-based techniques for sentence selection were disappointing compared to other methods, such as those leveraging sentence location and/or cue words and phrases (such as *"The purpose of this paper ....", "In summary ..."*, and so forth) [9] because frequency alone is a poor indicator of salience of terms, even when the stop words are ignored. More indicative is the *inverse document frequency* technique, adapted from information retrieval (proposed by [5] in the context of summarization in particular, it follows [36]), in which the relative frequency of an item in the document is compared with its relative frequency in a background collection.

The sentence selection process is based on a notion of salience; the most salient sentences identified are extracted for the summary. The *salience score of a sentence* is derived partly from the salience of vocabulary items (including single-token words, multi-word names, abbreviations, and multi-word terms, but excluding stop words) in the document and partly from its position in the document structure (e.g. section-initial, paragraph-internal, and so forth) and the salience of the surrounding sentences. The vocabulary items from the document are looked up in the collection vocabulary database by a statistical component that calculates, for each item, its inverse document frequency. This calculation compares the relative frequency of each item t in the document with the relative frequency of the item in the collection. This inverse document frequency measure is the item's salience score.

$$Salience(t) = \log_2 \frac{N_C / freq(t)_C}{N_D / freq(t)_D}$$

Salient items (*signature terms*, after [5]) are the items occurring more than once in the document, whose salience score is above an experimentally determined cutoff, or appearing in a strategic position in the document structure (e.g. title, headings, etc.). All other items are assigned zero salience.

The score for a sentence is made up of two components. The *salience* component is the sum of the salience scores of the items in the sentence. The *structure* component reflects the sentence's proximity to the beginning of the paragraph, and its paragraph's proximity to the beginning and/ or end of the document. Structure score is secondary to salience score; sentences with no salient items get no structure score. Still, a low- or non-scoring sentence might be selected anyway: thus sentences that immediately precede higher scoring ones in a paragraph may get promoted by virtue of an 'agglomeration rule', the operation of which is controllable from the client interface. Agglomeration addresses the problem of coherence discussed earlier (see Section 1); it is an inexpensive way of preventing dangling anaphors without having to identify them.

Another problem for sentence-based summarizers, also discussed in Section 1 above, is that of thematic under-representation (or, loosely speaking, coverage). This is addressed by another rule, the 'empty section' rule, which is of particular interest for this paper. Longer documents with multiple sections marked with headings, or news digests containing multiple stories may be unevenly represented in a sentence-extracted summary. The 'empty section' rule aims to ensure that each section is represented in the summary by forcing inclusion of its highest scoring sentences, or, if all sentence scores are zero, its first sentence.

In general, there are some exclusions to the sentence selection process. For example, sentences are excluded if they are too short (five words or less) or if they contain direct quotes (more than a minimum number of words enclosed in quotation marks)[4].

The summarization component described here performs best on documents within a certain genre: in effect, it assumes input of the type and length of a news story or news feature story (article). Furthermore, the requirement for a database of background statistics is clearly a crucial part of its design. This raises the two questions which are the point of departure for this paper. The first is how to handle situations where the input documents are longer, possibly significantly so, than the average length of a news story. The second concerns summarization of documents for which no background collection exists. Clearly, neither of these situations is extraordinary. It is easy to conceive of document collections in a different genre: scientific articles, patent descriptions, financial reports, and so forth, all exhibit length significantly beyond what the current summarizer is designed to represent. Furthermore, new documents are created all the time; by definition, these do not belong to any background collection. It may take time to accumulate such a collection and analyze it; it may be impractical to store the vocabulary statistics of such a collection; it may be the case that existing collections do not adequately reflect the domain and genre of new documents.

We have chosen to address these two questions by making the summarizer aware of certain discourse-level features of the document by leveraging the topic shifts in it; to this end, the TEXTRACT infrastructure has been augmented with a function for linear discourse segmentation.

[4] Note that as a result of the document structure constructed for each source text, such considerations are trivial to implement.

2.3. Discourse segmentation component

Our long term goal is to bring a degree of discourse awareness into the summarization process. Our approach is to make extensive use of *lexical cohesion.*

Discourse segmentation is driven by the determination of points in the narrative where perceptible discontinuities in the text cohesion are detected. Such discontinuities are indicative of topic shifts. Following the original idea of [24], subsequently developed specifically for the purposes of segmentation of expository text [13], we have adapted an algorithm for discourse segmentation to our document processing environment. In particular, while remaining sensitive to the distribution of "terms" across the document, and calculating similarity between adjacent text blocks by a cosine measure, our procedure differs from that in [13] in several ways.

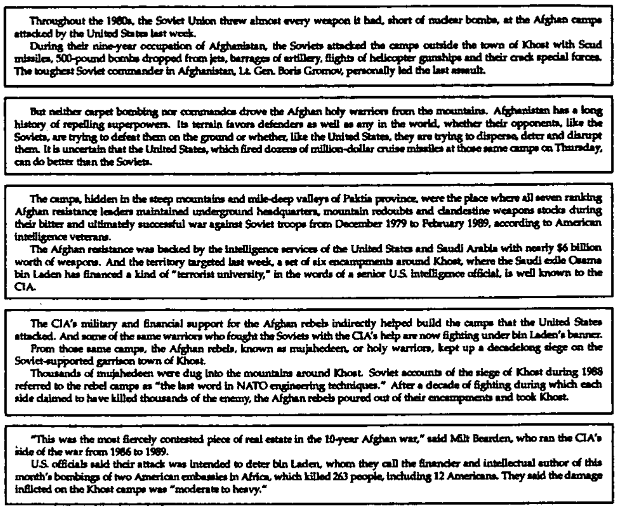
We only take into account content words (as opposed to all terms yielded by a tokenization step). These are normalized to lemma forms. "Termhood" is additionally refined to account for multi-word sequences (proper names, technical terms, and so forth, as discussed in Section 2.1 above), as well as some (limited) notion of co-reference, where different name variants get "aggregated" into the same canonical form ([39]). The cohesion calculation function is biased towards different types of possible break points: thus certain cue phrases (*"However"*, *"On the other hand"*) unambiguously signal a topic shift; document structure elements—such as sentence beginnings, paragraph openers, and section heads—are exploited for their 'pre-disposition' to act as likely segment boundaries; and so forth (see Section 2.1). The function is also adjusted to reduce the noise from block comparisons where the block boundary—and thus a potential topic shift—falls at unnatural break points (such as the middle of a sentence).

Modulo the above adjustments and modifications, we use essentially the same formula as Hearst's for computing lexical similarity between adjacent blocks of text b_1 and b_2 (t denotes a discourse element term identified as such by TEXTRACT's prior processing, ranging over the text span of the currently analyzed block; ω_{t,b_N}) is the normalized frequency of occurrence of the term in block b_N):

$$sim(b1, b2) = \Sigma_t \omega_{t,b_1} \omega_{t,b_2} / \sqrt{\Sigma_t \omega_{t,b_1}^2 \, \Sigma_t \omega_{t,b_2}^2}$$

In essence, we are able to utilize, transparently, the results of processes such as *lexical and morphological lookup*, *document structure identification*, and *cue phrase detection*, because these are already integral parts of our document processing environment (TEXTRACT). Likewise, the results of the segmentation process are naturally incorporated in an annotation superstructure which records the various levels of document analysis: discourse segments are just another type of a "span" over a number of sentences, logically akin to a paragraph.

Figure 1 illustrates the results of the 'raw' segmentation process.



Throughout the 1980s, the Soviet Union threw almost every weapon it had, short of nuclear bombs, at the Afghan camps attacked by the United States last week.

During their nine-year occupation of Afghanistan, the Soviets attacked the camps outside the town of Khost with Scud missiles, 500-pound bombs dropped from jets, barrages of artillery, flights of helicopter gunships and their crack special forces. The toughest Soviet commander in Afghanistan, Lt. Gen. Boris Gromov, personally led the last assault.

But neither carpet bombing nor commandos drove the Afghan holy warriors from the mountains. Afghanistan has a long history of repelling superpowers. Its terrain favors defenders as well as any in the world, whether their opponents, like the Soviets, are trying to defeat them on the ground or whether, like the United States, they are trying to disperse, deter and disrupt them. It is uncertain that the United States, which fired dozens of million-dollar cruise missiles at those same camps on Thursday, can do better than the Soviets.

The camps, hidden in the steep mountains and mile-deep valleys of Paktia province, were the place where all seven ranking Afghan resistance leaders maintained underground headquarters, mountain redoubts and clandestine weapons stocks during their bitter and ultimately successful war against Soviet troops from December 1979 to February 1989, according to American intelligence veterans.

The Afghan resistance was backed by the intelligence services of the United States and Saudi Arabia with nearly $6 billion worth of weapons. And the territory targeted last week, a set of six encampments around Khost, where the Saudi exile Osama bin Laden has financed a kind of "terrorist university," in the words of a senior U.S. intelligence official, is well known to the CIA.

The CIA's military and financial support for the Afghan rebels indirectly helped build the camps that the United States attacked. And some of the same warriors who fought the Soviets with the CIA's help are now fighting under bin Laden's banner.

From those same camps, the Afghan rebels, known as mujahedeen, or holy warriors, kept up a decadelong siege on the Soviet-supported garrison town of Khost.

Thousands of mujahedeen were dug into the mountains around Khost. Soviet accounts of the siege of Khost during 1988 referred to the rebel camps as "the last word in NATO engineering techniques." After a decade of fighting during which each side claimed to have killed thousands of the enemy, the Afghan rebels poured out of their encampments and took Khost.

"This was the most fiercely contested piece of real estate in the 10-year Afghan war," said Milt Bearden, who ran the CIA's side of the war from 1986 to 1989.

U.S. officials said their attack was intended to deter bin Laden, whom they call the financier and intellectual author of this month's bombings of two American embassies in Africa, which killed 263 people, including 12 Americans. They said the damage inflicted on the Khost camps was "moderate to heavy."

Figure 1: 'Raw' discourse segmentation: topic shifts

Informally, as a 'gloss' on the illustration above, the foci of the five segments could be described as:

- ❑ *Afghan camps thwart Soviets;*
- ❑ *Afghanistan history in repelling superpowers;*
- ❑ *Afghan resistance and US/Arab intelligence;*
- ❑ *Afghan rebels, and the siege of Khost;*
- ❑ *Target: Osama bin Laden.*

Most other applications of segmentation, typically in information retrieval, are primarily concerned with identifying segment boundaries: [14], [37], [30], [3], [35]. We are additionally interested in leveraging the content of the segments, to the extent that it is indicative of the focus of attention, and (indirectly, at least) points at the topical shifts which we need to utilize for the summary generation.

While it is unrealistic to expect that this kind of 'summary' could be automatically generated, it is our intent to use the segmentation results (together with the name and term identification and salience calculation delivered by other parts of TEXTRACT) in order to make sure that all the base data for inferring the topic stamps, and topic shifts, is available to the user.

This raises two related questions. The first concerns the relationship between segmentation and summarization: is segmentation a strictly "under the covers", service, function used by the summarizer, or might the results of discourse segmentation be of any interest, and use, to the end user? Unlike [17] (whose work also seeks to leverage linear segmentation for the explicit purposes of document summarization), we take the view that with an appropriate interface metaphor, where the user has an overview of the relationships between a summary sentence, the key salient phrases within it, and its enclosing discourse segment, a sequence of visually demarkated segments can impart a lot of information directly leading to the formulation of glosses like the one illustrated earlier. The second question thus concerns the features of such an interface. We return to this point later.

3. Discourse-aware summarization

As discussed in Section 1.1 above, common intuitions suggest a number of strategies for leveraging the results of linear discourse segmentation for enhancing summarization. In our testbed environment, we arranged for segmentation to 'publish' the topic shift points in the text into the document structure, by defining a segment as an additional type of document span (not dissimilar to sentence, paragraph, section, and so forth), with its own `from` and `to` coordinates; the summarizer thus transparently, and immediately, became aware of the segmentation results. We further arranged for a mechanism whereby certain strategies for incorporating segmentation results into the summarization process were easy to cast in summarizer terms. Thus, for instance, a heuristic which would require that each segment is represented in the summary is naturally expressed by treating segments as sections, and strictly enforcing the 'empty section' rule (see 2.2); a strategy which requires the selection of a segment-initial sentence for the summary is enabled simply by boosting the salience score for that sentence above a known threshold; a decision to drop an anecdotal segment from consideration in summary generation would be realised by setting, as a last step prior to summary generation, the sentence salience scores for all sentences in the segment to zeros.

For evaluating the effect of various strategies upon summarizer output quality, we used as baseline an evaluation corpus of full-length articles, and their 'digests', from *The New York Times*. There are advantages, and disadvantages, to this approach. Setting aside the issue of whether task-based evaluation (see below) is the appropriate mode for testing strictly the effect of one technology on another (see below, Section 3.1), such a decision ties us to a particular set of data. On the positive side, this offers a realistic baseline against which to compare strategies and heuristics; on the negative side, if a certain type of data is missing from the evaluation corpus, there is little hard evidence for judging the effects of strategies and heuristics on such data. In our particular case, even though an aspect of our investigation focused specifically at adequately summarizing long documents, the absence of such documents from the corpus prevents us from doing quantitative comparisons between summarizer output without, and with, segmentation.

At the time of writing, we are working with a customer organization with a need for summarizing long documents; we hope to be able to report the results of task-based evaluation *in situ* in due course. In the remainder of this section we focus on presenting the results for small-to-average size documents (the collection comprises just over 800 texts, less than half of which are over 10K, and virtually none are over 20K; the byte count includes HTML markup tags, in terms of number of sentences per document, very few of these longer documents are over 100 sentences long). First, we describe the evaluation environment.

3.1. Summarization evaluation testbed

Evaluating summarization results is not trivial. There is evidence that the optimal extract is not unique [32], [8]. The purpose of the extract varies; so do human extractors. Sentence extraction systems may be evaluated by comparing the extract with sentences selected by human subjects [32], [10], a (superficial) objective measure that ignores the possibility of multiple right answers. Another objective measure compares summaries with pre-existing abstracts using a suitable method for mapping a sentence in the abstract to its counterpart in the document [19]. Subjective measures, even though still less satisfying, can also be devised: for instance, summary acceptability has been proposed as one such measure [5]. Other evaluation protocols share the primary feature of being *task-based*, even though details may vary: performance may be measured by comparing browsing and search time as summary abstracts and full-length originals are being used [22], [38]; recall and precision in document retrieval [5]; or recall, precision, and time required in document categorization (i.e. assessing whether a document has been correctly judged to be relevant or not, on the basis of its summary alone) [11], [12].

During the development of the base summarization function in TEXTRACT, we built an environment for baseline evaluation 'in-house', as part of the development/training cycle. This same environment was used in analyzing the impact of discourse segmentation on the summarizer's performance. A background collection vocabulary statistics were gathered from analyzing 2334 *New York Times* news stories. Sentences in digests for 808 news stories and feature articles were automatically matched with their corresponding sentences in the full-length documents using a version of LINGUINI, a vector-based language identification program [31] that was able to map source to digest sentences even when slight differences existed between the two. Digests range in length from 1 to 4 sentences. Since we were particularly interested in longer stories, as well as stories in which the first sentence in the document did not appear in the digest, their representation in the test set, 38%, is larger than their distribution in the newspaper.

A limitation of this inexpensive test approach is the inherently short length of the digests, which prevents us from evaluating segmentation effects on summarization of long documents. Nonetheless, a number of comparative analyses can be carried out against this baseline collection, which are indicative of the interplay of the various control options, environment settings, and TEXTRACT filters used. One parameter, in particular, is quite instrumental in tuning the summarizer's performance, to a large extent because it is directly related to length of the original document: size of the summary, expressed either as number of sentences, or as percentage of the full length of the original. In addition to a clear intuition—size of the summary ought to be related to the size of the original—varying the length of the summary offers both the ability to measure the summarizer's performance against baseline summaries (i.e. our collection of digests), and the potential of dynamically adjusting the derived summary size to optimally represent the full document content, depending on the size of that document.

We conducted our experiments with different granularities of summary size. In principle, the performance of a system which does absolute sentence ranking, and systematically picks the N 'best' sentences for the summary, should not depend on the summary size. In our case, the additional heuristics for improving the coherence, readability, and representativeness of the summary (see Section 2.2) introduce variations in overall summary quality, depending on the compaction factor applied to the original document size. A representative spectrum for the test corpus we use is given by data points at: digest size (i.e. summary exactly the size, expressed as number of sentences, of the digest); 4 sentences; 10% of the size of the full length document; and 20% of the document. Not surprisingly (for a salience-based system), the summarization function alone, without discourse segmentation, benefits from larger summary size. Although the recall rate is higher still for longer summaries, it is not a measure of the overall quality of the summary because of the inherently short length of the digest.

3.2. Segmentation effects on summarization

Elaborating the intuitions outlined in Section 1.1, our experiments compare the base summarization procedure, calculating object salience with respect to a background document collection (Section 2.2), with enhanced procedures incorporating several different strategies for leveraging the notions of discourse segments and topic shifts.

The experiments fall in either of two categories. In an environment where a background collection, and statistics, cannot be assumed, a summarization procedure was defined to take selected (typically initial) sentences from each segment; this appeals to the intuition that segment-initial sentences would be good topic indicators for their respective segments. The other category of experiment focused on enriching the base summarization procedure with a sentence selection mechanism which is informed by segment boundary identification and topic shift detection.

In combining different sentence selection mechanisms, several variables need adjustment to account for relative contributions of the different document analysis methods, especially where summaries can be specified to be of different lengths. Given the additional sentence selection factors interacting with absolute sentence ranking, we again set the granularity of summary size at three discrete steps, mirroring the evaluation of the original summarizer: summaries can be requested to be precisely 4 sentences long, or to reflect source compaction factor of 10% or 20% (Section 3.1).

In general, we experimented with two strategies for actively incorporating topical information into the summary: one was to add the segment-initial sentences to the set of sentences already selected by the salience calculation mechanism, the other was to exert finer control over the number of sentences selected via salience, and 'pad' the summary to its requested size with sentences selected from segments by invoking the 'empty segment' (aka 'empty section', see 2.2) rule. Special provisions were made to account for the fact that segmentation would naturally always select the first sentence in the document.

It turns out that the differences between a range of realisations of the above two strategies are not statistically significant over our test corpus; we thus use the label "SUM+SEG" to denote a 'composite' strategy and to represent the whole family of variations. In contrast, "SUM" refers to the base summarization component, and "SEG" represents summarization by segmentation alone. Table 1 below shows the recall rates for the three major summarization regimes defined by different summary granularities. Since segmentation effects are clearly very different across different sizes of source document, our experiments were additionally conducted at sampling the document collection at different sizes of the originals: the corpus was split into four sections, grouping together documents less than 7.5K characters long, 7.5–10K, 10–19K, and over 19K;for brevity, the table encapsulates a 'composite' result (denoted by the label *"All documents"*). What is of particular interest here is that the complete set of data from these experiments makes it possible, for any given document, to select dynamically the summarization strategy appropriate to its size, in order to get an optimal summary for it, in any given information compaction regime.

	4 sents	10%	20%
All documents			
SEG	54.74	54.74	56.09
SUM	46.85	49.71	66.47
SUM+SEG	56.52	56.30	58.37
All documents with > 1 digest sentence			
SEG	45.13	45.13	46.78
SUM	36.34	39.84	58.66
SUM+SEG	41.64	46.75	51.65
All documents whose 1st sentence is not in target digest			
SEG	31.12	32.73	33.99
SUM	29.93	39.96	61.71
SUM+SEG	32.53	41.45	47.96

Table 1: Summary data for segmentation effects

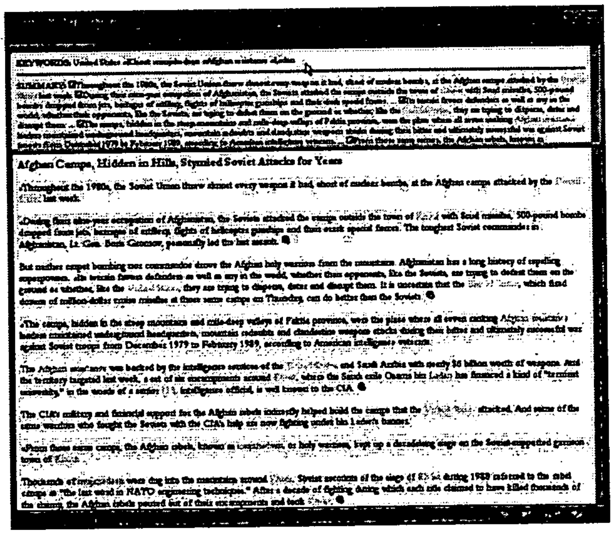
In order to get a better sense for the effects of different strategy mixes, we show results for the same summarization regimes, on subsets of the test corpus. *"All documents with > 1 digest sentence"* represents documents whose digests are longer than a single sentence; *"All documents whose 1st sent is not in target digest"* extracts a document set for which a baseline strategy automatically picking a representative sentence for inclusion in the summary would be inappropriate. These subset selection criteria explain the deterioration of overall results; however, what is more interesting to observe in the table is the relative performance of the three summarization regimes.

Overall, leveraging some of the segmentation analysis is positively beneficial to summarization; the effects are particularly strong where short summaries are required. In addition, the summarization procedure defined to work from segmentation data alone shows recall rates comparable to, and in certain situations even higher than, the original TEXTRACT function: this suggests that such a procedure is certainly usable in situations where background collection-based salience calculation is impossible, or impractical.

4. Seeing the topics shift

Unlike other TEXTRACT functions, which act like linguistic filters, and typically are incorporated 'under the hood' in larger systems (such as query expansion in information retrieval [6], or document navigation in knowledge management [27]), the summarization component stands alone. The user sees, directly, the result of summarizing a document; Figure 2 illustrates a typical view of a document and its summary.

Without going into details (see [26]), the major characteristic of this interface is that the two windows, the summary one at the top and the original document at the bottom, are asynchronously controlled via separate scroll bars. This is far from satisfactory, primarily because it makes it difficult to use the summary as a navigation tool into the complete document content. Various heuristics have been proposed to alleviate this problem, most of them using the notion of hyperlinking summary sentences with their counterparts in the full document [20], and the interface illustrated here (Figure 2) employs a similar contextualisation device [27].



KEYWORDS: [illegible]

SUMMARY: [illegible]

Afghan Camps, Hidden in Hills, Stymied Soviet Attacks for Years

Throughout the 1980s, the Soviet Union threw almost every weapon it had, short of nuclear bombs, at the Afghan camps attacked by the [illegible] last week.

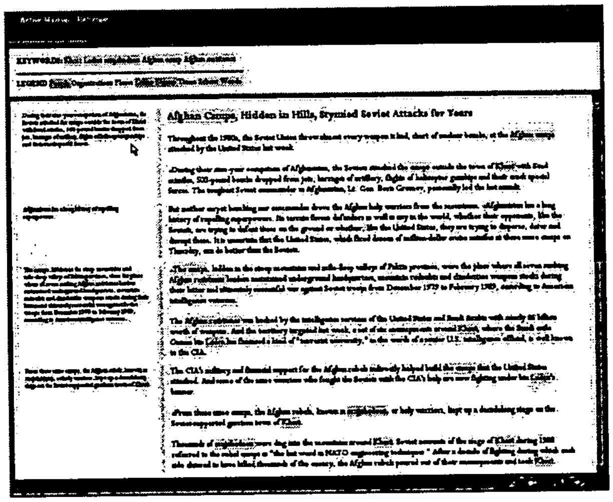
[illegible]

Figure 2: TEXTRACT summarizer: early view

This is suboptimal, primarily because the jumps from a sentence in the summary to its position in the document are abrupt, and because there are no visual indicators to suggest how two adjacent summary sentences relate to each other (if at all) in the document. There may be arbitrary amount of material intervening, which has been omitted from the summary: knowing this, as well as knowing the extent of the span of the missing material, is essential for better understanding the summary [4]. In effect, the only way of making some sense of the summary as an abstraction of the full document is by being able to 'undo' the effects of ellision of material between each two adjacent summary sentences.

Representing the fragments missing from the source is very hard to arrange for by means of a visual abstraction, because, as a direct consequence of the problem of under-representation (Section 1) in the canonical summarization framework, the client typically has no control over the extent of the material which falls below the sentence salience threshold. However, since discourse segmentation is intended to address this problem, it also turns out to offer the means of a richer visual abstraction, which directly incorporates the notion of topic shifts at the interface. We thus take the view that segmentation is not only a subsidiary function for enhancing the quality of summarization, but a process which is of independent utility for the end user, as long as its results are integrated within an appropriate interface.[5]

Figure 3 presents a screen snapshot of a prototype front end to a segmentation-enhanced summarizer, which is capable of contextualising summary sentences, indicating the span of omitted material between them, and suggesting grouping of summary fragments to show topic highlighting. A crucial feature of this interface is that the two different information panes, the summary one on the left and the full length document on the right, are synchronously scrollable; furthermore, both displays are 'anchored' to the segment span visual abstraction—the vertical bar in the middle—which is the primary organizational device mediating both the results of the summary sentence selection and topic shift detection.



KEYWORDS: [illegible]

LEGEND: [illegible]

Afghan Camps, Hidden in Hills, Stymied Soviet Attacks for Years

Throughout the 1980s, the Soviet Union threw almost every weapon it had, short of nuclear bombs, at the Afghan camps attacked by the United States last week.

[illegible]

Figure 3: TEXTRACT summarizer: segmentation overlay

[5] By way of informally defining the notion of "appropriate", it is worth noting that the representation in Figure 1 is not an appropriate end-user visualization of discourse segmentation.

It is worth noting that without discourse segmentation, this kind of visual metaphor would be very hard to render on a summary stream which does not have topical information in it (such as illustrated in Figure 2). Due to the under-representation problem, the summary (left) pane might be too sparse; visually, this would translate into mis-cueing the user whether what is seen in the summary pane is a complete summary, or a fragment whose continuation is only reachable after (arbitrary amount of) scrolling. Additional problems arise from lack of any data to facilitate the user in identifying topics missing from the summary in what would be a long passage in the right pane, without any topical (or other) annotation.

The interface makes use of additional features: a hyperlink device to facilitate attention switching between the summary and document panes, while retaining focus on a topical sentence; colour coding for marking and displaying salient vocabulary items; hot spots to highlight recurring occurrences of a salient item. We will not discuss these in detail here, as they are not directly related to the integration of segmentation and summarization functions (but see [26]).

5. Conclusion

We have addressed a class of problems inherent to summarization-by-sentence-extraction technology, by developing a discourse segmentation component capable of detecting shifts in topic, and integrating this within a linguistically-aware summarizer which utilizes notions of salience (with respect to a background document collection) and dynamically-adjustable size of the resulting summaries. By analyzing coherence indicators in the discourse, segmentation identifies points in the narrative where substories alternate; these are used to define for the summarization function a set of discourse segments, the representation of which makes for more complete, informative and faithful to the original summaries.

Under certain conditions, segmentation-enhanced summarization is better than the base segmentation technology utilized in TEXTRACT. Some of these conditions can be expressed as a function of the original document length, and the document-to-summary ratio: this makes it possible to select the optimal strategy for combining the two technologies 'on the fly'.

In addition, having access to a segmentation component makes it possible to alleviate a serious shortcoming of the TEXTRACT summarizer: in situations where background collection-based salience calculation is impossible, or impractical, it is still possible to deliver summaries generated by access to discourse segmentation information alone. These have been shown to be of comparable quality, yet considerably cheaper to generate.

This work is part of a larger effort focused on leveraging elements of the discourse structure in an attempt to recognize and use cohesive devices in text for a variety of content characterisation tasks. Additional interesting extensions within the same space of functional enhancements would lead to augmenting the base-level segmentation component with a simple measure of 'connectedness' between any two discourse segments; thus, by picking different chains of cohesively connected segments, different perspectives on the document content could be revealed; by dynamically adjusting the threshold of acceptably connected segments, summaries of different length can be generated. The hope is that, in either case, the resulting summaries would display higher degree of cohesion than that of a sequence of sentences, due to the thematically (more) complete nature of the discourse segments, which are the basic unit for content mediation in the new summaries. We are currently working on the infrastructure for deeper cohesion analysis.

We are also experimenting with more dynamic interfaces, capable of fully utilizing the results of multiple analyses, both in the context of single document summaries, and content-mediated navigation in a document collection. Extensions and modifications to current interface metaphors incorporte notions like larger (and guaranteed to be thematically coherent) text fragments, representative sentences which may be more or less central/peripheral to a given summary thread, multiple threads (summaries) through the same document source, and multi-level document abstractions mediated via different levels of granularity of content.

References

[1] AAAI 1998 Spring Symposium Series. *Intelligent Text Summarization (Working Papers)*, Stanford, CA, March 1998.

[2] C. Aone, M. E. Okurowski, J. Gorlinsky, and B. Larsen. A scalable summarization system using robust NLP. In *Intelligent Scalable Text Summarization, Proceedings of a Workshop Sponsored by the Association for Computational Linguistics*, pages 66–73, 1997.

[3] D. Beeferman, A. Berger, and J. Lafferty. Text segmentation using exponential models. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997.

[4] B. Boguraev, R. Bellamy, and C. Kennedy. Dynamic presentations of phrasally-based document abstractions. In *Hawaii International Conference on System Sciences (*HICSS-32*): Understanding Digital Documents*, Maui, Hawaii, January 1999.

[5] R. Brandow, K. Mitze, and L. Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5):675–685, 1995.

[6] R. Byrd, Y. Ravin, and J. Prager. Lexical assistance at the information-retrieval user interface. In *Proceedings of the 4th International Symposium on Document Analysis and Information Retrieval*, pages 485–493, Las Vegas, NV, 1995.

[7] D. Caruso. New software summarizes documents. The New York Times, January 27 1997.

[8] F. R. Chen and M. M. Withgott. The use of emphasis to automatically summarize a spoken discourse. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 229–232, 1992.

[9] H. Edmundson. New methods in automatic abstracting. *Journal of the ACM*, 16(2):264–285, April 1969.

[10] H. Edmundson. New methods in automatic abstracting. *Journal of the ACM*, 16(2):264–285, 1969.

[11] T. F. Hand. A proposal for task-based evaluation of text summarization systems. In *Intelligent Scalable Text Summarization, Proceedings of a Workshop Sponsored by the Association for Computational Linguistics*, pages 31–38, 1997.

[12] T. F. Hand and B. Sundheim, editors. TIPSTER/SUMMAC *Summarization Analysis; Tipster Phase* III *18-Month Meeting”.*, NIST, Fairfax, Virginia, 1998. Defense Advanced Research Project Agency. Working papers from SUMMAC conference.

[13] M. Hearst. Multi-paragraph segmentation of expository text. In *32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, 1994.

[14] M. A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, CO, 1995.

[15] F. Johnson, C. Paice, W. Black, and A. Neal. The application of linguistic processing to automatic abstract generation. *Journal of Documentation and Text Management*, 1(3):215–241, 1993.

[16] J. S. Justeson and S. M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27, 1995.

[17] M.-Y. Kan, J. L. Klavans, and K. R. McKeown. Linear segmentation and segment significance. In E. Charniak, editor, *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 197–205, Montreal, Canada, August 1998. Sponsored by ACL and ACL's SIGDAT.

[18] C. Kennedy and B. Boguraev. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of COLING-96 (16th International Conference on Computational Linguistics)*, Copenhagen, DK, 1996.

[19] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, Seattle, Washington, 1995.

[20] K. Mahesh. Hypertext summary extraction for fast document browsing. In *Proceedings of AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, pages 95–104, Stanford, CA, 1997.

[21] I. Mani, T. Firmin, and B. Sundheim. The TIPSTER SUMMAC text summarization evaluation. In *Proceedings of the Ninth Conference of the European Chapter of the ACL*, pages 77–85, Bergen, Norway, June 1999. Association for Computational Linguistics.

[22] S. Miike, E. Itho, K. Ono, and K. Sumita. A full text retrieval system with a dynamic abstract generation function. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 152–161, 1994.

[23] M. Mitra, A. Singhal, and C. Buckley. Automatic text summarisation by paragraph extraction. In I. Mani and M. Maybury, editors, *Proceedings of a Workshop on Intelligent Scalable text Summarization*, pages 39–46, Madrid, Spain, 1997. Sponsored by the Association for Computational Linguistics.

[24] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–48, 1991.

[25] M. S. Neff. A system for text summarization by sentence extraction. Technical report, IBM T.J. Watson Research Center, 1999. In preparation.

[26] M. S. Neff and J. W. Cooper. ASHRAM: active summarization and markup. In *Hawaii International Conference on System Sciences (*HICSS-32*): Understanding Digital Documents*, Maui, Hawaii, January 1999.

[27] M. S. Neff and J. W. Cooper. A knowledge management prototype. In *Proceedings of the 4th International Conference on Applications of Natural Language to Information Systems*, Klagenfurt, Austria, June 1999.

[28] C. Paice and P. Jones. The identification of important concepts in highly structured technical papers. In R. Korfhage, E. Rasmussen, and P. Willet, editors, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 69–78. ACM Press, 1993.

[29] C. D. Paice. Constructing literature abstracts by computer: techniques and prospects. *Information Processing and Management*, 26:171–186, 1990.

[30] J. Ponte and W. Croft. Text segmentation by topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 120–129, 1997.

[31] J. Prager. LINGUINI: language identification for multilingual documents. In *Hawaii International Conference on System Sciences (*HICSS-32*): Understanding Digital Documents*, Maui, Hawaii, January 1999.

[32] C. Rath, A. Resnick, and T. Savage. The formation of abstracts by the selection of sentences. *American Documentation*, 12(2):139–143, 1961.

[33] Y. Ravin and Z. Kazi. Is Hillary Rodham Clinton the President? disambiguating names across documents. In *Workshop on Coreference and Its Applications*, Maryland, June 1999. Association for Computational Linguistics.

[34] Y. Ravin and N. Wacholder. Extracting names from natural-language text. Technical Report RC 20338, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY, 1997.

[35] J. Reynar. *Topic segmentation: algorithms and applications.* PhD thesis, University of Pennsylvania, Department of Computer and Information Science, 1998.

[36] G. Salton and M. McGill. *An Introduction to Modern Information Retrieval.* McGraw-Hill, 1993.

[37] G. Salton, A. Singhal, C. Buckley, and M. Mitra. Automatic text decomposition using text segments and text themes. In *Seventh ACM Conference on Hypertext*, Washington, D.C., 1996.

[38] K. Sumita, K. Ono, and S. Miike. Document structure extraction for interactive document retrieval systems. In *Proceedings of SIGDOC*, pages 301–310, 1993.

[39] N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of names in text. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 202–208, Washington ,D.C., March 1997.